APPLICATION FOR UNITED STATES LETTERS PATENT

For

SYSTEM FOR PROVIDING A MULTIMEDIA PEER-TO-PEER COMPUTING PLATFORM

Inventor(s):

Yen-Kuang Chen

Matthew J. Holliman

Li-Cheng Tai

Rainer W. Lienhart

Igor V. Kozintsev

Minerva M. Yeung

Prepared by:

BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP
1279 Oakmead Parkway
Sunnyvale, California 94086-4039
(408) 720-8300

Attorney's Docket No.: 042390.P11076

"Express Mail" mailing label number: EL 617 209 426 US

Date of Deposit: June 8, 2001

I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has been addressed to the Commissioner for Patents, Washington, D. C. 20231

Yolanda Kepner

(Typed or printed name of person mailing paper or fee)

(Signature of person mailing paper or fee)

(Date signed)

SYSTEM FOR PROVIDING A MULTIMEDIA PEER-TO-PEER COMPUTING PLATFORM

BACKGROUND

[0001] Peer-to-peer scenarios may be exemplified by the absence of a "server" in a traditional client-server environment. Such a paradigm may be viewed as an instance of distributed computing, where a system of (often heterogeneous) nodes operate in a cooperative or confederated fashion to complete a given task. A number of examples have been recently reported, including hybrid approaches such as that of Napster, which uses a centralized client-server model for lookups and direct peer-to-peer communication for file transfers, as well as completely distributed file sharing architectures such as Gnutella.

[0002] In general, peer-to-peer applications thus contrast to a traditional client's dependence on a server. In reality, however, the distinction between peer-to-peer and client-server is often blurred, and peer nodes can be viewed as taking on the roles of both a client and a server.

[0003] Current peer-to-peer infrastructures, however, typically do not provide flexible/dynamic support for operations such as multimedia services. For example, typical peer-to-peer infrastructures implement "download"-style data transfers via Transmission Control Protocol (TCP) only, despite the fact that multimedia can be streamed and viewed in real time using an unreliable protocol such as User Datagram Protocol (UDP). Compounding this problem, some peer-to-peer infrastructures do not

even support direct source-to-destination transfers, instead propagating data node-by-node through the network.

[0004] Moreover, many current systems limit the scope of user queries to queries by filename, making it difficult for users to devise innovative new query mechanisms that can take advantage of the richer semantics of media content. Finally, multimedia files exhibit different characteristics than other files. Multimedia files can exist in a variety of different formats, resolutions, and qualities. Hence, depending on the end user's requirements, a multimedia-aware delivery system should be able to send the same in different format to different users.

[0005] As such, there is a need to support a more dynamic peer-to-peer platform for exchanging diverse data such as multi-media data.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] **Figure 1** is a block diagram of the interoperation of peers within a Global Network Universe according to one embodiment.

[0007] **Figure 2** is a block diagram illustrating the support module according to one embodiment.

[0008] **Figure 3** is a block diagram of the interoperation of peers within a Global Network Universe according to one embodiment.

[0009] **Figure 4** is a flow diagram describing the operations according to one embodiment.

DETAILED DESCRIPTION

[0010] The present application describes a peer-to-peer (P2P) support module to improve sharing of data/resources in a peer-to-peer environment. Illustrated in Figure 1 is a P2P network configuration of peers (e.g., nodes, computers, set-top boxes, ect.) One or more of the nodes within the network configuration may include a P2P support module of the present invention to improve the sharing of data/resources. As further illustrated in Figure 1, node 120 includes a P2P support module, according to one embodiment, in the memory of the node.

[0011] One characteristic of a Peer-to-peer environment is transparency of the physical location of a resource. That is, the location of data and other resources need not be determined by the applications using the peer-to-peer network. The data or resources (e.g., computation/Central Processing Unit (CPU) bandwidth, storage, etc.) can exist anywhere in the peer-to-peer global network universe, and the system handles the discovery and the delivery/utilization of the data/resource to the requesting application of a peer node. To support the transparency, in one embodiment, the P2P support module is separated from the traditional Operating System, as illustrated in more detail in Figure 2. The P2P support module interfaces with the network transport services 104 of the operating system 102, according to one embodiment.

[0012] The P2P support module, in one embodiment as illustrated in node 120, includes P2P service layer 106 (described in greater detail below). In addition, the P2P support module includes support handlers 108, which provide transformation and computation services on the data/resources obtained via the peer-to-peer service layer

106. The handlers assist in making the system modular, flexible and extensible. For example, in one embodiment, the support modules may provide transcoding (format conversion) support (e.g., MPEG 2 to MPEG 4). In addition, the support modules may also provide multimedia watermarking, and other security features.

[0013] In addition, a scripting language interpreter 110, may also be provided to support applications 112 with programmatic access to the services provided by the P2P support module below.

[0014] In alternative embodiment, additional components/services may be provided and/or some of the components/services discussed above may not be included, without departing from the present invention.

[0015] A subset of the P2P support module, Tthe peer-to-peer service layer 106, supports providing the illusion of data location transparency throughout a network of nodes via the following features discussed in detail.

Transport Mechanisms

[0016] In a peer-to-peer environment, data is likely to be transmitted over a variety of heterogeneous communication medium including telephone lines, high-speed wired networks, wireless local area networks, Bluetooth networks, and mobile cellular networks etc. Typically, in existing peer-to-peer networks, the transport protocols used are typically reliable in nature. While this approach masks the specifics of the underlying channel and is amenable to rapid prototyping and implementation, it may not be well

suited for real-time delivery of multimedia data. Further, even in the case of delay-insensitive media data, wireless peers are likely to have limited storage resources and the concurrent playback and streaming of the data may thus be limited.

[0017] As such, considering the variety of different network conditions and setups on the Internet, a flexible peer-to-peer framework should support different transport mechanisms. In one embodiment, the network transport interface 104 between the peer-to-peer Service Layer 106 and the OS 102, is hot-pluggable. In particular, the network transport interface supports the incorporation of different transport handlers (e.g., TCP, RTP/UDP, etc.). As a result, in one embodiment, the peer support module allows the transport mechanism to be specified by the application, chosen from the set of available protocols supported by the network transport services.

[0018] For example, a video player application on a wireless handheld device might request to the peer services layer 106 that it only receive a video via RTP/UDP in combination with Forward Error Correction (FEC) and Automatic Repeat Request methods, in order to support real-time viewing of the stream. In contrast, a delay-insensitive application that prefetches and downloads videos on to a home server might specify the use of a simpler default protocol such as TCP as its transport protocol. The peer services layer 106 of the peer support module obtains the requested data at the source host(s) via the necessary protocol(s), pursuant to the request of the application.

[0019] Furthermore, in an alternative embodiment, the peer services layer provides support for wireless devices. In particular, the peer services layer is aware of the status of the wireless peer, such as battery state, location, mobility, etc., in addition to parameters common to the other peers (e.g. available bandwidth, screen resolution, available media players, etc). In one embodiment, wireless peers can announce their capabilities and status, for other peers to better accommodate media delivery to them. For example, if a wireless peer signals a low battery state, the other peers react by changing the media transport protocol for this wireless peer to be more power efficient (usually at the expense of reducing, e.g., media quality).

[0020] Furthermore, in an alternative embodiment, the application support module may adjust the personalization of content delivered to the device depending on its (device) location. For example, in a museum/exhibition a wireless peer device may receive access to the media content, which is linked to a particular item located in the vicinity on the peer device. Another example is having media data delivered to both the mobile device and another device (e.g., speakers, standalone monitor, display wall) that is located in the vicinity of a mobile device for synchronized delivery of multimedia data.

Data Naming

[0021] In one embodiment, the peer-to-peer service layer 106 of the P2P support module supports user-defined representations of data. In one embodiment, the P2P service layer 106 may map user-defined names or metadata to a fixed Globally Unique

Identifier (GUID), which may then be used by all peers to identify a piece of data and optionally track different copies of the data around the network.

[0022] In one embodiment, peer nodes throughout the network include directory services, which provide a view of where data is stored in the network. In one embodiment, the directory is a hash table mapping user-assigned names to the GUIDs, although in alternative embodiments, the directory may also support a tree structure or other organization structures. The directory services of different peer nodes may work in a cooperative fashion to help establish a view of the global network universe for each peer node. In another embodiment, which may be appropriate for relatively smaller systems requiring fast directory service lookups, global directory information may be duplicated across nodes.

[0023] In one embodiment of the P2P support module, user-specified query/resource handlers are provided in the peer nodes, to support customized queries better suited to individual content types. In this approach, in one embodiment, a user-specified query/resource handler in a peer node is used to locate a resource matching a given query. For example, a query for "video stream with mountains in the background" may be routed to a user-specified query/resource handler, which can use peer daemon Application Program Interfaces (APIs) to locate the requested resources.

[0024] In the case of directory service lookups, in one embodiment a peer daemon invokes the same query on other peers that support the same query mechanism. In an

alternative embodiment, in addition to supporting user-defined queries, the system may provide a number of pre-defined query types that are available on all peers, e.g. to locate resources by filename, by file type, etc. Furthermore, in one embodiment of the system, peers on the network may cache the results to previous queries.

Dynamic resource retrieval (transcoding)

[0025] In one embodiment, locating a version of a resource on the network is determined by a computation of a user-defined cost function (i.e., a metric that attempts to rank and prioritize available paths to the resource). In this context, a path consists of both a conventional network route and/or sequence of transformations throughout the network that are necessary to deliver the requested resource to the requesting node.

[0026] In order to deliver the requested version of a resource, existing copies of the requested resource may need to be transformed by the support modules 108 to satisfy constraints of the application/peer node requesting the resource. For example, in the case of requested video, existing content in MPEG-2 format may need to be downscaled spatially and temporally, quantized, and converted to a low bitrate MPEG-4 stream for viewing on a wireless device.

[0027] In another example, a user without access to Microsoft® PowerPoint® may request a version of a presentation as a GIF file only, which could result in dynamic transformation of the content to the necessary format. Such transcoding may be accomplished on demand on sufficiently powerful machines, or in another embodiment

as the result of background processes operating on host nodes, which may perform speculative content adaptation, e.g. based on previous requests.

[0028] In one embodiment, the cost function and constraints imposed by the user would determine whether a new copy of the content would be generated, targeted specifically for the specific requirements, or whether an existing copy would suffice. By minimizing the computed cost function, the peer service layer determines the most appropriate technique for computing and obtaining the transformed resource or accessing a pre-stored version.

[0029] In addition, in one embodiment, the data to be transmitted could be transcoded based on the network bandwidth/error rate between peer nodes. In one embodiment, the data could be transcode into a low-bitrate content when network is congested and cannot handle a higher-bitrate streaming in real-time. In another embodiment, the data could be transcoded into a more error-resilient format when the communication link is wireless and the channel coding quality is low.

System Implementation

[0030] In one embodiment, as illustrated in Figure 3, the peer-to-peer nodes each include a peer-to-peer daemon 220 and a client API 222 to be linked into applications using the peer-to-peer service. The peer-to-peer daemon is responsible for connecting with the peer-to-peer daemons at other peer-to-peer nodes to form neighboring relationships and to transmit queries/data/resources.

[0031] In one embodiment, the peer-to-peer daemon provides support for peer-to-peer operations. The daemon handles the transport of metadata, actual data, and query requests.

[0032] In one embodiment, the basic unit of traffic on the peer-to-peer network is a *packet*. One embodiment of a packet contains the following fields:

1. Command: the type or the purpose of the request contained in the packet.
2. Arguments: the list of arguments specific to the command.
3. Auxiliary information for the packet (e.g., GUID, made time, expiration time, sequence number, time to live value, a cost value, and ID of the node that send this request).

[0033] In one embodiment, the following example commands are used as described in the method described in the flow diagram of Figure 4. In step 402 NAME GUID is announced. The NAME is a user-provided meta- description of a resource, and the GUID is the system-assigned id for the resource. This command informs neighboring peer nodes of the presence of a new resource.

[0034] In step 404, a query SEARCH-STRING is issued by a node searching for data/resource. The SEARCH-STRING is a user-specified query searching for data/resources. In one embodiment, each node interprets the query as it sees fit; and the requesting node does not specify a standard query language or query semantics.

[0035] For example, in one embodiment, a request for media data consist of two parts: a description of the media data requested as well as, a description of the format in which the media data should be delivered.An example of a video delivery format description is: width = 320; height = 240; bit rate = 512kbits/s; type = mpeg4.

[0036] For retrieval purpose, media data is described by metadata such as the time of creation, place of creation, content of the data, etc. In one embodiment, each media data has one special metadata field called GUID. The GUID assumes that each media data is unique at its insertion time into the peer-to-peer network, and therefore a globally unique ID should be assigned to it. Replicas (with the same binary representation or in a transcoded representation) will have the same GUID in order to show that they all represent the same 'unique' media data.

[0037] An example of a metadata description for videos is: title = "The Matrix"; Content = "Meeting Morphius"; or GUID = 87245792438952394-SDGK-3453.[MH1][ik2]

[0038] In step 406, an answer SEARCH-STRING GUID is sent as a reply in response to the query requests. The resource that satisfies the query SEARCH-STRING is identified by GUID. In one embodiment, the command only answers queries and informs of the existence of resources but does not provide the actual data. In one embodiment, the GUID of the resource may change as the ANSWER packet passes through a separate peer node. More specifically, when the data is actually forwarded

from the separate subsequent node, the data with an old GUID will be converted to a new copy with a different GUID representing the separate subsequent node.

[0039] In step 408, a Get GUID requesting a resource identified by GUID is sent by the node requesting the resource/data. In step 410, a Put GUID DATA [expiration-time] command is sent actually carrying the DATA or the content resource identified by GUID. For a PUT packet, the GUID may change and the data may get transformed to a new format at a separate subsequent node as the packet is transmitted throughout the peer network.

[0040] In one embodiment, a cost value inside each packet is used when passing an ANSWER packet. In particular, each node has a cost function that adds a cost value to the cost of the request. The cost function contains the cost function computation handler, which can be installed dynamically by the user and takes two arguments: the request data specification and the real data specification. Given the specifications, the handler computes the actual cost of doing data conversion from the real data specification to the request data specification at this node, if the response node has the capability to do the conversion. The cost value is then added to the Answer packet. The answers are presented to the original querying client who can then choose one data resource/GUID.

[0041] In one embodiment, if a local node cannot provide the desired information, it forwards the requests to all its neighbors, minus the node where the request originally comes from. Moreover, in one embodiment, the answer command leaves a record in the nodes along the path of its return to the original querying node. As a result, in one

embodiment, each node may keep a record of the GUID of the Answer, the GUID of the answered resource, its metadata, and the GUID of a new copy of the resource if the node supports data conversion and has trancoded/converted the resource.

[0042] Thus when a Get command arrives at a node the node may proceed to check its records of past Answers and may change the GUID argument of the Get packet to the incoming GUID of the past Answer packet. Similarly, when the requested resource identified by the incoming GUID actually arrives, the peer node may invoke an external data conversion program to create the resource identified by the outgoing GUID and then put the new resource in the Put packet before forwarding it.

[0043] The architecture and methods described above can be stored in the memory of a computer system (e.g., set top box, video recorders, etc.) as a set of instructions to be executed. In addition, the instructions to perform the operations described above could alternatively be stored on other forms of machine-readable media, including magnetic and optical disks. For example, the operations of the present invention could be stored on machine-readable media, such as magnetic disks or optical disks, which are accessible via a disk drive (or computer-readable medium drive). Further, the instructions can be downloaded into a computing device over a data network in a form of compiled and linked version.

[0044] Alternatively, the logic to perform the operations as discussed above, could be implemented in additional computer and/or machine readable media, such as discrete

hardware components as large-scale integrated circuits (LSI's), application-specific integrated circuits (ASIC's), firmware such as electrically erasable programmable read-only memory (EEPROM's); and electrical, optical, acoustical and other forms of propagated signals (e.g., carrier waves, infrared signals, digital signals, etc.); etc.

[0045] Although the present invention has been described with reference to specific exemplary embodiments, it will be evident that various modifications and changes may be made to these embodiments without departing from the broader spirit and scope of the invention. Accordingly, the specification and drawings are to be regarded in an illustrative rather than a restrictive sense.